Pertemuan 1

Pengenalan Java

Objektif:

1. Mahasiswa mengetahui latar belakang perkembangan bahasa Java.
2. Mahasiswa mengetahui fitur-fitur penting yang terdapat pada Java.
3. Mahasiswa mengerti konsep dan struktur bahasa pemrograman Java.
4. Mahasiswa mengerti konsep variabel, tipe data, dan operator pada Java.

P1.1 Teori

1. Latar Belakang Bahasa Java

1.1 Sejarah Singkat Java

Pada tahun 1991, sekelompok insinyur SUN yang dipimpin Patrick Naughton dan James Gosling ingin merancang bahasa komputer untuk perangkat consumer seperti cable Tv box. Karena perangkat itu tidak mempunyai banyak memori, bahasa harus berukuran kecil dan menghasilkan kode yang liat, maka bahasa harus bebas dari arsitektur manapun. Proyek ini diberi nama kode Green.

Produk pertama proyek Green adalah “*7”, sebuah kendali jauh yang sangat cerdas. Pada tahun 1995, Netscape memutuskan membuat browser yang dilengkapi dengan Java. Setelah itu diikuti IBM, Symantec, Inspire, bahkan Microsoft. Bahasa Java merupakan karya Sun Microsystem Inc. Rilis resmi dilakukan pada Nopember 1995. Nama JAVA sendiri terinspirasi pada saat orang–orang di proyek Green sedang menikmati secangkir kopi di sebuah kedai kopi yang kemudian dengan tidak sengaja salah satu dari mereka menyebutkan kata JAVA yang mengandung arti asal bijih kopi. Maskot Java adalah *Duke.*

1.2 Pengenalan Teknologi JAVA

- **Java sebagai Sebuah Bahasa Pemrograman**

  Java adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai platform sistem operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai system operasi dan bersifat open source.
- **Java sebagai Sebuah Development Environment**

  Sebagai sebuah peralatan pembangun, teknologi Java menyediakan banyak tools : compiler, interpreter, penyusun dokumentasi, paket kelas dan sebagainya.
- **Java sebagai Sebuah Aplikasi**

  Aplikasi dengan teknologi Java secara umum adalah aplikasi serba guna yang dapat dijalankan pada seluruh mesin yang memiliki Java Runtime Environment (JRE).
- **Java sebagai Sebuah Deployment Environment**

  Terdapat dua komponen utama dari Deployment Environment. Yang pertama adalah JRE, yang terdapat pada paket J2SDK, mengandung kelas–kelas untuk

semua paketteknologi Java yang meliputi kelas dasar dari Java, komponen GUI dan sebagainya. Komponen yang lain terdapat pada Web Browser. Hampir seluruh Web Browser komersial menyediakan interpreter dan runtime environment dari teknologi Java.

1.3 Pembagian Edisi Paket pada JAVA

Untuk beragam aplikasi yang dibuat dengan bahasa Java, Java dipaketkan dalam edisi-edisi berikut:

1. *Java 2 Standar Edition (J2SE)*, J2SE menyediakan lingkungan pengembangan yang kaya fitur, stabil, aman, dan cross-platform. Edisi ini mendukung konektivitas basis data, rancangan user interface, masukkan/ keluaran (input/output), dan pemrograman jaringan (network programming), dan termasuk sebagai paket-paket dasar bahasa Java.
2. *Java 2 Enterpise Edition (J2EE)*, J2EE menyediakan tempat untuk membangun dan menjalankan multitier enterprise editions. J2EE berisi paket-paket J2SE serta paket-paket untuk mendukung pengembangan Enterprise JavaBeans, Java Servlets, JavaServer Pages, XML, dan kendali transaksi yang fleksibel.
3. *Java 2 Micro Edition (J2ME)*, J2ME selain menyedikan bahasa Java yang sama, unggul dalam portabilitas (kemampuan dapat dijalankan dimanapun), *safe network delivery*, seperti J2SE dan J2EE. Aplikasi-aplikasi dapat diskalakan (dimampukan) agar dapat bekerja dengan J2SE dan J2EE. J2ME adalah untuk beragam *consumer electronic produc*t, seperti *pager, smart card, cell phone, handheld PDA*, dan *set-top box*.

1.4 Karakteristik JAVA

Java memiliki karakteristik sebagai berikut :

✓ Sederhana

Bahasa pemrograman Java menggunakan sintaks mirip dengan C++ namun sintaks pada Java telah banyak diperbaiki terutama menghilangkan penggunaan pointer yang rumit dan multiple inheritance. Java juga menggunakan automatic memory allocation dan memory garbage collection.

✓ Berorientasi objek (Object Oriented)

Java mengunakan pemrograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali. Pemrograman

berorientasi objek memodelkan dunia nyata kedalam objek dan melakukan interaksi antar objek-objek tersebut.

- ✓ Dapat didistribusi dengan mudah

  Java dibuat untuk membuat aplikasi terdistribusi secara mudah dengan adanya libraries networking yang terintegrasi pada Java.

- ✓ Interpreter

  Program Java dijalankan menggunakan interpreter yaitu Java Virtual Machine (JVM). Hal ini menyebabkan source code Java yang telah dikompilasi menjadi Java bytecodes dapat dijalankan pada platform yang berbeda-beda.

- ✓ Robust

  Java mempuyai reliabilitas yang tinggi. Compiler pada Java mempunyai kemampuan mendeteksi error secara lebih teliti dibandingkan bahasa pemrograman lain. Java mempunyai runtime-Exception handling untuk membantu mengatasi error pada pemrograman.

- ✓ Aman

  Sebagai bahasa pemrograman untuk aplikasi internet dan terdistribusi, Java memiliki beberapa mekanisme keamanan untuk menjaga aplikasi tidak digunakan untuk merusak sistem komputer yang menjalankan aplikasi tersebut.

- ✓ Architecture Neutral

  Program Java merupakan platform independent. Program cukup mempunyai satu buah versi yang dapat dijalankan pada platform yang berbeda dengan Java Virtual Machine.

- ✓ Portabel

  Source code maupun program Java dapat dengan mudah dibawa ke platform yang berbeda-beda tanpa harus dikompilasi ulang.

- ✓ Performance

  Performance pada Java sering dikatakan kurang tinggi. Namun performance Java dapat ditingkatkan menggunakan kompilasi Java lain seperti buatan Inprise,Microsoft ataupun Symantec yang menggunakan Just In Time Compilers (JIT).

2. Fitur-fitur Pada Java

2.1 Java Virtual Machine (JVM)

JVM adalah sebuah mesin imajiner (maya) yang bekerja dengan menyerupai aplikasi pada sebuah mesin nyata. JVM menyediakan spesifikasi hardware dan platform dimana kompilasi kode Java terjadi. Spesifikasi inilah yang membuat aplikasi berbasis Java menjadi bebas dari platform manapun karena proses kompilasi diselesaikan oleh JVM. Aplikasi program Java diciptakan dengan file teks berekstensi .java. Program ini dikompilasi menghasilkan satu berkas bytecode berekstensi .class atau lebih.

2.2 Garbage Collection

Program Java melakukan garbage collection yang berarti program tidak perlu menghapus sendiri objek–objek yang tidak digunakan lagi. Fasilitas ini mengurangi beban pengelolaan memori oleh programmer dan mengurangi atau mengeliminasi sumber kesalahan terbesar yang terdapat pada bahasa yang memungkinkan alokasi dinamis.

2.3 Code Security

Code Security terimplementasi pada Java melalui penggunaan Java Runtime Environment (JRE). Java menggunakan model pengamanan 3 lapis untuk melindungi sistem dari untrusted Java Code.

3 . Struktur Program Java

Dalam pembuatan program Java yang harus diperhatikan dalam pembuatan program java adalah penulisan huruf besar dan kecil karena java memiliki sifat *Case Sensitive*. Berikut adalah bentuk umum dari penulisan program Java :

Pertama dalam program Java minimal terdapat sebuah class, dimana nama dari class tersebut diusahakan sama dengan nama file Java, dan setiap class harus dibuka dengan tanda ‘{‘ dan ditutup dengan tanda ‘}’.

Contoh:
```
class coba {
    (isi dari class)
    }
```

Kedua dalam program Java terdapat fungsi main() adalah dijadikan sebagai awal pengeksekusian aplikasi Java, kode (code) yang terdapat pada metode inilah yang akan dieksekusi pertama kali.

Contoh:

```
class coba
{
public static void main(String[] args)
{
(tulis code/ program disini)
}
	}
```

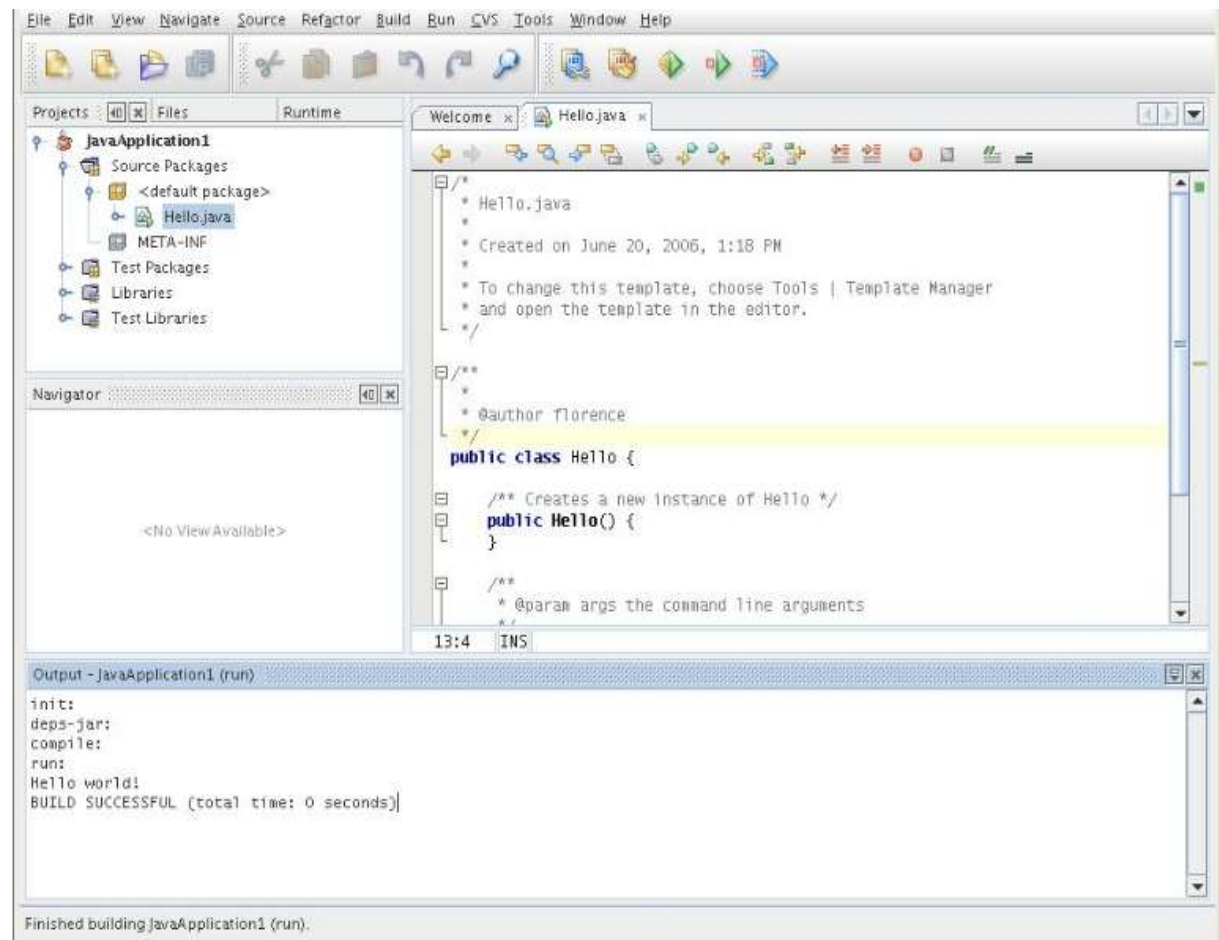
Metode main () didefinisikan sebagai public static void, berikut penjelasannya :

- ✓ public, berarti metode ini dapat dipanggil dari luar class
- ✓ static, menunjukkan metode ini bersifat sama untuk semua class
- ✓ void, berarti metode ini tidak mengembalikan nilai.
- ✓ Argument args [] adalah array objek string argument baris-baris perintah yang dilewatkan ke kelas yang di eksekusi.

<u>Contoh Program Sederhana JAVA</u>

```
// nama file : Hello.java
class Hello {
	public static void main(String[] args)
	   {
	System.out.println("Hello, World!");
	}
}
```

Pada program di atas, langkah pertama yang kita lakukan yaitu membuat class Java dengan nama Hello. Class tersebut memiliki fungsi main() dengan parameter variabel args dengan tipe array dari String. Fungsi main ini yang akan mengeksekusi statement System.out.println (“Hello World!”); untuk menampilkan kata Hello World!

File tersebut disimpan dengan akhiran ekstensi .java yang kemudian akan dijalankan di Netbeans sehingga muncul output seperti gambar di bawah ini.

4. Variabel, Tipe Data dan Operator pada Java

4.1 Variabel pada Java

Variabel digunakan dalam program untuk menyimpan suatu nilai, dan nilai tersebut dapat diubah selama eksekusi program. Sedangkan konstanta menyatakan nilai yang tidak berubah pada saat eksekusi.

Bentuk Umum Deklarasi Variabel yaitu :

[tipe data] [nama variabel] [;]

Contoh :

```
int harga;
char nama hari;
double bonus;
```

Berikut ini aturan-aturan dalam membuat variabel pada Java :

- ✓ Case sensitive, huruf kecil dan kapital berbeda. Contoh: alamat, Alamat, dan ALAMAT menyatakan tiga variabel yang berbeda.
- ✓ Nama dari sebuah variabel harus dimulai dengan sebuah huruf dan selanjutnya dapat diikuti dengan huruf atau angka.
- ✓ Huruf yang bisa digunakan adalah dari 'A'-'Z', 'a'-'z', angka dari 0-9 dan juga simbol '_' dan ‘$’ , Untuk simbol lain dan spasi tidak dapat digunakan.
- ✓ Dalam penamaan variabel juga tidak diperkenankan menggunakan reserved word atau keyword dari Java.
- ✓ Panjang variabel tidak dibatasi.

  Dalam pendeklarasian variabel kita juga bisa menyatukan dua variabel dalam pendeklarasiannya

  Contoh :

  ```
  int i, j;
  long y,x;
  char a,b;
  ```

4.2 Tipe Data Pada Java

Tipe data mendefinisikan metode penyimpanan untuk mereperesentasikan informasi dan cara informasi diinterprentasikan. Tipe data berkaitan erat dengan penyimpanan variabel di memori karena tipe data variabel menentukan cara kompilator menginterpretasikan isi memori. Tipe data dalam Java dibagi 2 kategori:

1. Sederhana / primitive, terdapat 8 tipe-tipe sederhana dan dipisahkan dalam 4 kelompok:
   - ➢ Empat tipe adalah untuk bilangan bulat (integer) bertanda: byte, short, int, dan long.
   - ➢ Dua untuk tipe angka titik mengambang (floating point) atau bilangan pecahan: float dan double.
   - ➢ Satu untuk tipe karakter yaitu char, mewakili simbol pada himpunan karakter seperti tulisan dan angka.
   - ➢ Satu untuk tipe Boolean, merupakan tipe khusus untuk menunjukkan besaran logika (nilai-nilai logika).
2. Komposit, tipe data komposit disusun dari tipe data sederhana atau tipe komposit lain yang telah ada. Tipe ini antara lain: string, array, class, dan interface.

4.3 Operator Pada Java

Ekspresi adalah statement yang mengembalikan suatu nilai sedangkan Operator adalah suatu symbol yang biasanya digunakan dalam ekspresi. Operator akan melakukan suatu operasi terhadap operand sesuai dengan fungsinya, seperti penjumlahan, pengurangan, perkalian, dan sebagainya. Operator di Java ada 7 macam, yaitu :

1. Operator penugasan
2. Operator matematika
3. Operator penaikan dan penurunan
4. Operator bitwise (manipulasi bit)
5. Operator relasional
6. Operator logika Boolean
7. Operator majemuk

1. Operator Penugasan

Operator ini adalah operator yang paling sederhana dan hanya dilambangkan dengan karakter '='. Operator ini digunakan untuk menugaskan suatu nilai ke suatu variabel.

Ekspresi	Arti
x += y	x = x + y
x -= y	X = x – y
x *= y	x = x * y
x /= y	x = x / y

2. Operator Aritmatika

Operator ini digunakan dalam ekspresi matematika. Berikut ini daftar operator matematika yang dikenal dalam Java.

Operator	Arti	Contoh
+	Penjumlahan	3 + 4
-	Pengurangan	5 – 7
*****	Perkalian	5 * 5
/	Pembagian	14 / 7
%	Modulus	20 % 7

3. Operator Perbandingan / Relasional

Operator ini digunakan untuk membandingkan dua buah nilai operand dan menghasilkan nilai Boolean yaitu true atau false. Berikut ini operator relasional dalam Java.

Operator	**Keterangan**	**Contoh**
==	Sama dengan	x== 3
!=	Tidak sama dengan	x != 3
<	Lebih kecil dari	x < 3
>	Lebih besar dari	x > 3
<=	Lebih kecil dari atau sama dengan	x < = 3
>=	Lebih besar dari atau sama dengan	x > = 3

4. Operator Penaikan dan Penurunan

Operator ini digunakan untuk menaikkan atau menurunkan suatu nilai integer (bilangan bulat) sebanyak satu satuan, dan hanya digunakan pada variabel.

Operator	Keterangan
++	Penambahan dengan nilai satu
--	Pengurangan dengan nilai satu

Sebagai contoh :

```
x = x + 1;
y = y -1;
```

dapat ditulis menjadi:

```
x++ ;
y-- ;
atau
++x ;
--y ;
```

5. Operator Bitwise (manipulasi bit)

Operator ini digunakan untuk operasi secara langsung terhadap bit yang menyusun suatu bilangan. Daftar operator bitwise dalam Java adalah sebagai berikut :

<table>
<tr><th>Operator</th><th>Arti</th></tr>
<tr><td>-</td><td>Bitwise negasi (NOT)</td></tr>
<tr><td>&</td><td>Bitwise AND</td></tr>
<tr><td>|</td><td>Bitwise OR</td></tr>
<tr><td>^</td><td>Bitwise XOR</td></tr>
<tr><td>>></td><td>Geser kanan</td></tr>
<tr><td>>>></td><td>Geser kanan tanpa mempertahankan sign (dengan nilai 0 sebagai pengisi bit paling kiri)</td></tr>
<tr><td><<</td><td>Geser kiri</td></tr>
</table>

Karena operasi bitwise memanipulasi hingga ke bagian yang paling kecil dari suatu bilangan, yaitu bit maka diperlukan penjelasan lebih lanjut tentang bagaimana Java menyimpan reprsentasi suatu bilangan dalam bit, terutama sekali untuk representasi bilangan negatif.

Java menyimpan representasi dari suatu bilangan sebagai bilangan basis dua (binary) yang tersusun atas bit. Contoh bilangan binary : 000100111. angka 0 dan 1 yang membentuk bilangan binary inilah yang disebut bit.

6. Operator Logika Boolean

Operator ini hanya digunakan untuk melakukan operasi pada operand yang bertipe Boolean. Berikut daftar operator logika Boolean.

<table>
<tr><th>Operator</th><th>Keterangan</th></tr>
<tr><td>&</td><td>Logika AND</td></tr>
<tr><td>|</td><td>Logika OR</td></tr>
<tr><td>^</td><td>Logika XOR</td></tr>
<tr><td>!</td><td>Logika NOT</td></tr>
<tr><td>&&</td><td>Short Circuit Logika AND</td></tr>
<tr><td>|</td><td>Short Circuit Logika OR</td></tr>
<tr><td>==</td><td>Logika sama dengan</td></tr>
<tr><td>!=</td><td>Logika tidak sama dengan</td></tr>
</table>

7. Operator Majemuk

Operator ini merupakan gabungan dari operator yang telah dibahas sebelumnya dengan operator penugasan. Berikut operator majemuk pada Java.

Operator	Keterangan
+=	Penambahan diikuti dengan penugasan
- =	Pengurangan diikuti dengan penugasan
* =	Perkalian diikuti dengan penugasan
/ =	Pembagian diikuti dengan penugasan
% =	Operasi modulus diikuti dengan penugasan

Berikut ini contoh penggunaannya :

Hasil +=10 : memiliki arti yang sama dengan hasil = hasil+10

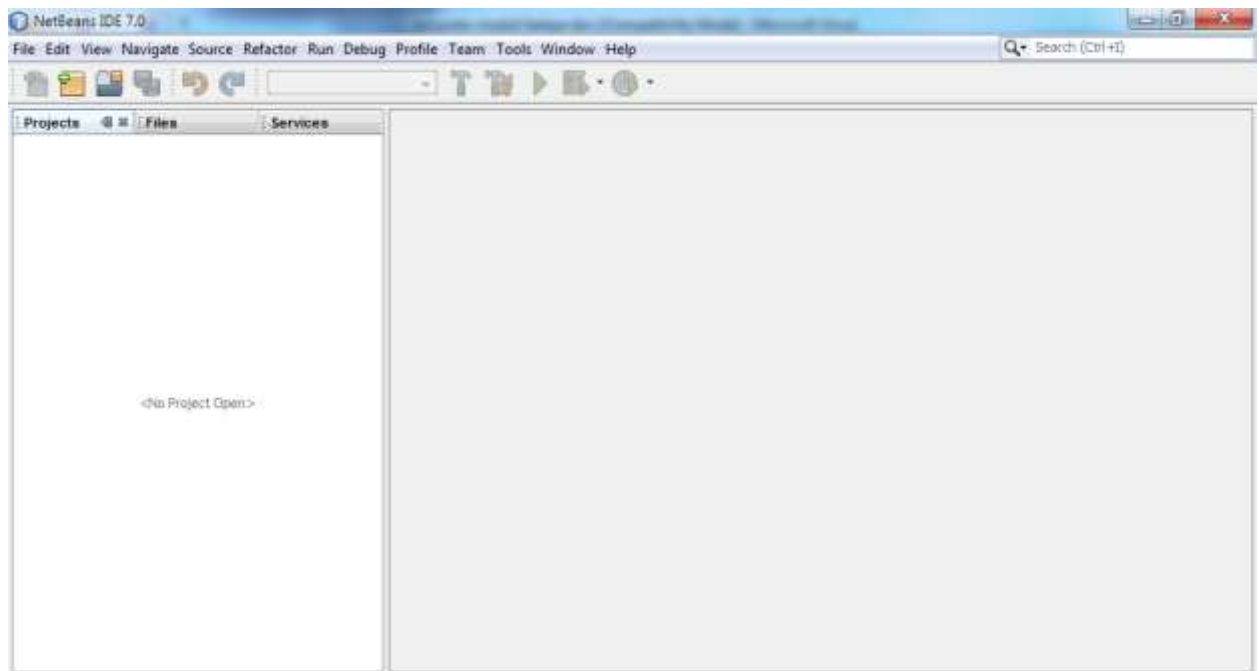
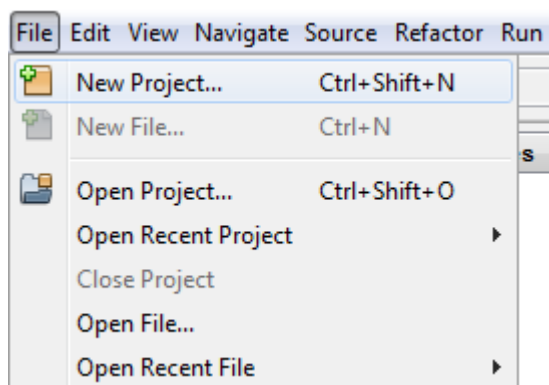
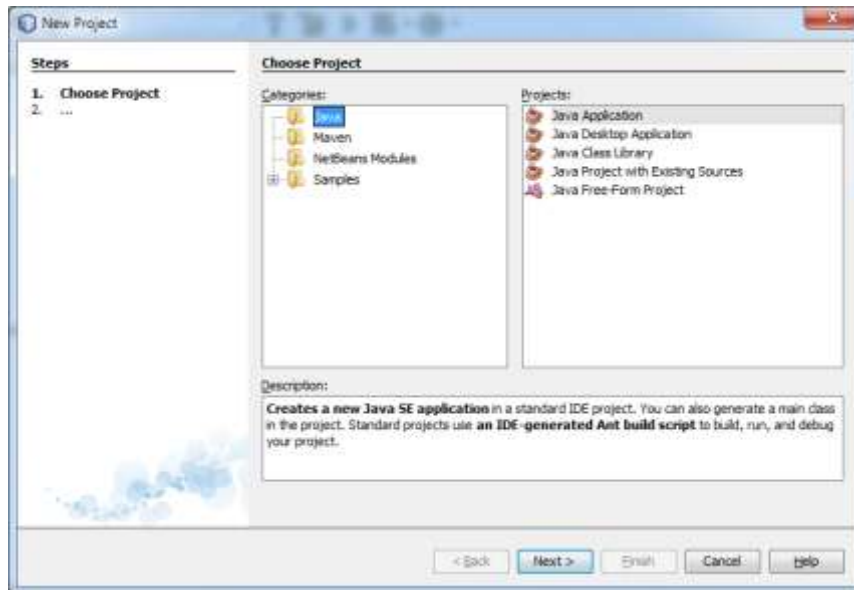
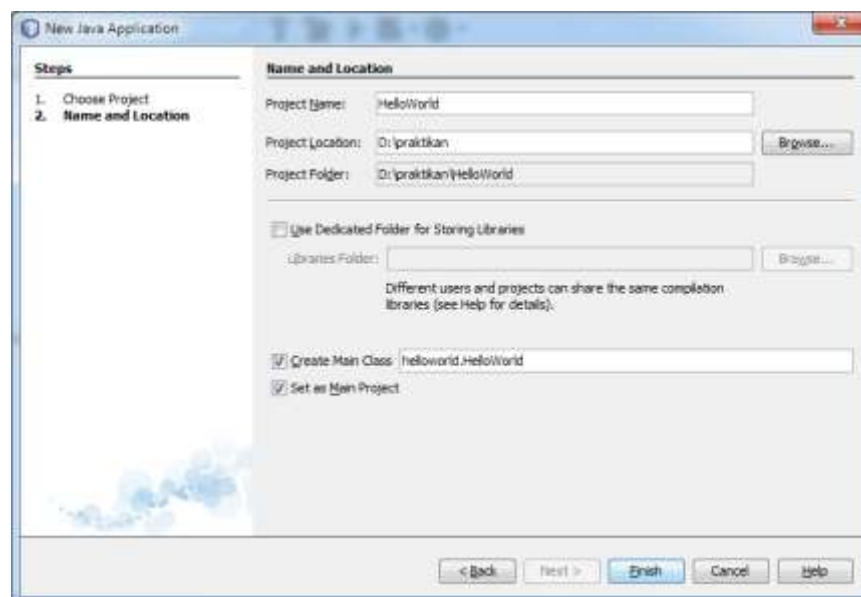
Hasil -=20 : memiliki arti yang sama dengan hasil = hasil-20

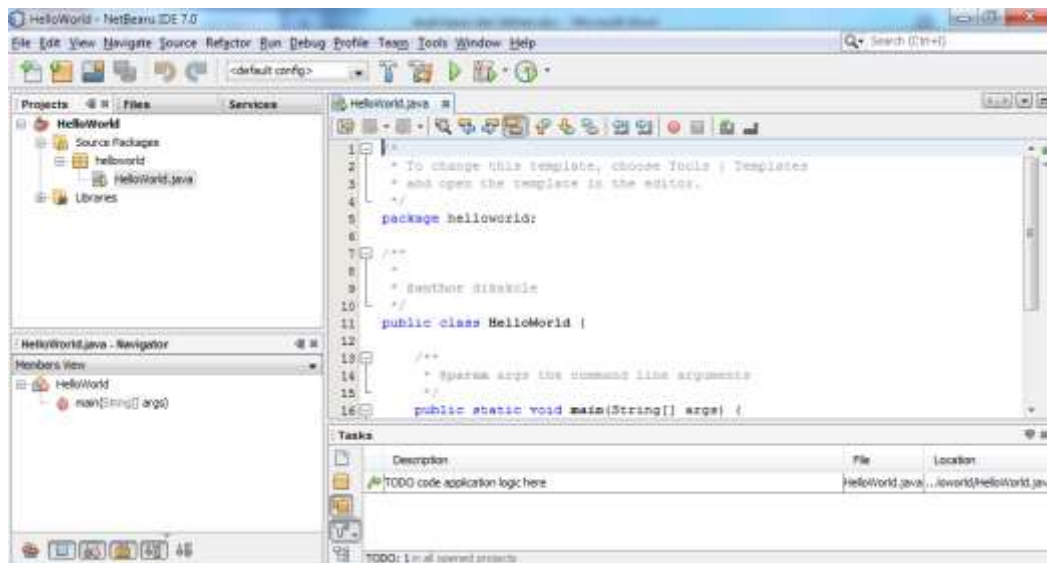
P1.2 Contoh Kasus

Seorang mahasiswa bernama Andika diminta untuk membuat program sederhana menggunakan bahasa pemrograman java dengan menggunakan editor Netbeans versi 7.0. Program yang dibuat akan menghasilkan output seperti pada gambar di bawah ini :

Output - belajarjava (run)
Tasks
run:
hello world
BUILD SUCCESSFUL (total time: 2 seconds)

Langkah-langkah Pengerjaan

1. Klik icon Netbeans 7.0 pada desktop, hingga muncul tampilan seperti pada gambar di bawah ini

2. Klik menu file dan pilih New Project

3. Pada New Project Wizard, klik kategori java dan pilih Java Application kemudian klik Next

4. Kemudian langkah berikutnya ialah :
   - Pada field **Project Name** ketikkan HelloWorld
   - Pada field **project Location**, secara default akan disimpan pada direktori yang telah ditentukan. Pada contoh ini yaitu direktori D:\praktikan

   - Pada field **Create Main Class** ketikkan helloworld.HelloWorld (pada Netbeans nama class mengikuti nama project yang kita ketikkan)
   - Kemudian set checkbox pada **Set as main project dan Create Main Class**

5. Klik Finish maka akan terlihat tampilan seperti berikut

- Terdapat project window untuk melihat struktur komponen project termasuk di dalamnya kode sumber yang dimasukkan dan library dari code yang akan dmasukkan
- Source editor window dengan file yang bernama Helloworld
- Navigator Window yang berguna untuk mengatur elemen dalam sebuah class/file yang di buat
- Task window, untuk melihat hasil output dari program yang kita buat atau melihat kesalahan dari program yang dibuat.

6. Untuk membuat source code yang ingin di buat, maka dapat di ketik pada project window di bawah komentar

```
// TODO code application logic here
```

Karena pada editor netbeans nama package dan class telah terdefinisi saat kita membuat sebuah project.

Kemudian setelah anda diminta untuk membuat output seperti pada kasus di atas maka anda harus menuliskan pemecahan kasusnya dengan sintaks

```
System.out.println("Hello World!");
```

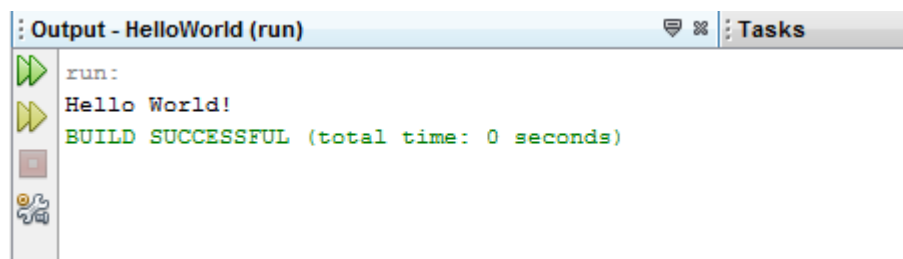
Perhatikan gambar di bawah ini untuk memperjelas tampilan project windownya.

```
package helloworld;

/**
 *
 * @author dikakole
 */
public class HelloWorld {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
            System.out.println("Hello World!");
    }
}
```

7. Lalu save file tersebut, pilih menu file dan pilih Save As atau dengan shortcut pada keyboard yaitu Ctrl+s.

8. Kemudian untuk meng-*compile* program yang dibuat, anda klik menu Run dan pilih run File atau dengan shortcut pada keyboard Shift+F6. Untuk mengcompile seluruh program dalam satu project HelloWorld dan dijadikan dalam bentuk Jar maka pilih menu Run dan pilih Build Main Project atau dengan shortcut pada keyboard F11.

9. Setelah itu jalankan program yang kita buat dengan memilih menu Run lalu pilih Run File untuk menjalankan output yang diminta. Seperti yang terlihat pada gambar di bawah ini.

```
Output - HelloWorld (run)                    Tasks
run:
Hello World!
BUILD SUCCESSFUL (total time: 0 seconds)
```

P1.3 Latihan

```
package helloworld;

/**
 *
 * @author dikakole
 */
public class math {
  public static void main (String []args){

      ……………….//deklarasikan variable I dengan nilai 10
      int j = 5 ;
      ………………..//deklarasikan variable K

     k = ++i + ++j;
     System.out.println("Nilai K adalah " + k);
  }

}
```

Pertanyaan :

1. Untuk mengisi titik-titik di atas, maka deklarasi yang benar untuk variable I adalah
   A. I = 10
   B. I = 10;
   C. int I = 10;
   D. int I := 10;
   E. int I = 10.
   Jawab: C

2. Sedangkan deklarasi yang benar untuk variable K adalah
   A. int K
   B. int K;
   C. num K
   D. num K;
   E. num K.
   Jawab: B

3. Apa Output yang ditampilkan dari program di atas ?
   A. K = 17
   B. Nilai K adalah 17
   C. Nilai K 17
   D. K adalah 17
   E. Nilai K = 17
   Jawab: B

P1.4 Daftar Pustaka

Naughton, Patrick, *Java Handbook: Konsep Dasar Pemrograman Java*, Andi Yogyakarta, 1996.

Gary Cornell dan Cay S.Horstmann, *Core Java edisi Indonesia*, Andi, Yogyakarta, 1997.

ANuff, *Penuntun Pemrograman Java*, Andi Yogyakarta, 1997.

Abdul Kadir, *Dasar Pemrograman Java 2*, Andi Yogyakarta, 2008.